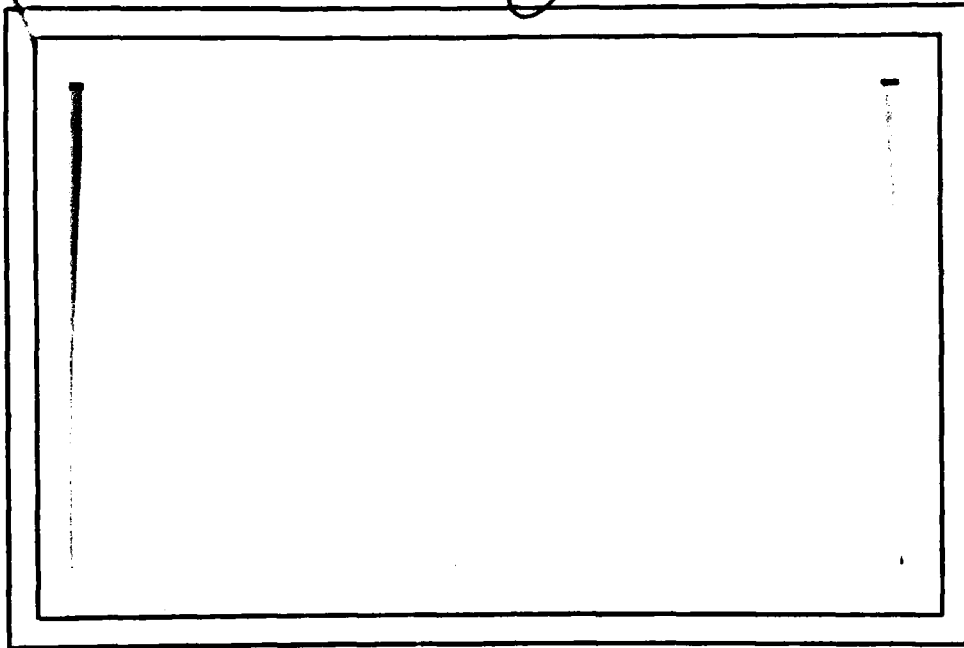


LEVEL

6.5
①

ADA 086086



COMPUTER SCIENCE
TECHNICAL REPORT SERIES



UNIVERSITY OF MARYLAND
COLLEGE PARK, MARYLAND

20742

DTIC
ELECTE
S JUL 2 1980
A

DDC FILE COPY

THIS DOCUMENT IS BEST QUALITY PRACTICAL.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

80 6 30 090

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

14 TR-845

11 December, 1979

15 DAAG-53-76C-0138

DARPA Order - 3296

6 A METHOD FOR FINDING PAIRS
OF ANTI-PARALLEL STRAIGHT LINES.

10 Ann Scher
Michael Shneier
Azriel Rosenfeld

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

9 Technical rept. series

ABSTRACT

A method of pairing anti-parallel straight lines is presented and discussed. The pairing is based on the distance between the lines, the amount by which they overlap, and on whether or not other lines are interposed. Examples are shown of applying the method to high-resolution aerial photographs. Results indicate that cultural features such as roads and buildings can be extracted and that a significant reduction in the complexity of the image description can be obtained.

DTIC
ELECTE
S JUL 2 1980 D
A

The support of the Defense Advanced Research Projects Agency and the U.S. Army Night Vision Laboratory under Contract DAAG-53-76C-0138 (DARPA Order 3206) is gratefully acknowledged, as is the help of Kathryn Riley in preparing this paper.

PM
411074

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

1. INTRODUCTION

Prepared For	WAS GMAI	Availability Codes
By	Unrecovered	
Distribution	Justification	Available/or special
Dist.		AP

A characteristic of cultural features that appear in high-resolution aerial photographs is the large number of straight edges. Usually, these edges occur in pairs, as in the sides of roads and of buildings. As a preliminary step towards identifying such features, the straight edges can be extracted (Nevatia and Babu, 1978; Peleg, 1978), and clustered into antiparallel pairs (i.e. pairs of facing edges that are parallel but have opposite directions). The clustering process is the subject of this report. In general, clustering must take into account information from the picture in the regions around the edges. For example, a road usually has a uniform gray level (Quam, 1978, Tavakoli, in progress) and thus it is reasonable to expect the facing sides of an antiparallel pair of edges to have a similar gray level. The process to be described here, however, deals only with a line drawing extracted from an image, and does not have information of this sort available to it. It has nevertheless proved itself a robust and reasonably successful technique, despite its relative simplicity.

The process accepts as input ordered pairs of endpoints defining lines and their directions. It attempts to find pairs of lines that are more or less antiparallel and that obey relations concerning overlap and distance between the lines. The process considers all lines that have similar orientations, and uses an iterative procedure to find good pairs. Whereas previous work (Nevatia and Babu, 1978, Brooks, 1979) has restricted the choice of pairs to lines that are closest neighbors, the current algorithm makes a more global analysis.

The basic procedure is as follows. Each line is compared with all other lines that are approximately antiparallel to it, i.e. those that lie within a length-dependent threshold. The amount of variation in the estimated direction of a line is

inversely proportional to the length of the line, so that short lines, whose directions are less certain, are allowed a greater discrepancy than longer lines. The figure of merit for a given pairing of lines is based on the amount of overlap between the lines, i.e. the proportion of common length of their projections, and the distance between the lines. Several ways of combining these measures were evaluated, and the results indicated that the exact combination was not critical to the procedure.

Having given initial scores to the pairs of lines, the next, iterative, step involves refining the scores by interactions among the lines that compete for the same antiparallel partner. Each line has a link to every possible antiparallel pairing line. Each link has a link strength, and the link strengths are normalized. Should a line have no links to other lines or should there be no links to it, the line is dropped from consideration. If two antiparallel lines A and B are a mutually best pair, i.e., if A's link strength to B is stronger than its link strength to any other line, and B's link to A is also its strongest link, and there are no lines between A and B, then A and B are considered to be linked and can be dropped from further consideration. Note that when this happens, any other lines that were linked to A or to B must have these links deleted. Because of the deletions a renormalization is required, and this adjusts the link strengths with respect to alternative anti-parallel pairs for lines. The process is iterated until no more pairs can be formed. Figure 1a shows a picture of part of an airfield. Figure 1b shows the set of directed edges derived from the picture, and Figure 1c shows the results of applying the process to the lines in Figure 1a.

The rest of this paper describes the method in detail, presents examples of its application to sets of lines extracted from aerial photographs, and discusses some of the shortcomings and advantages of the approach.

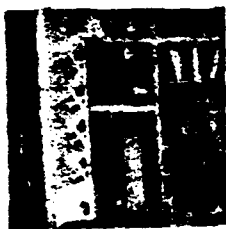
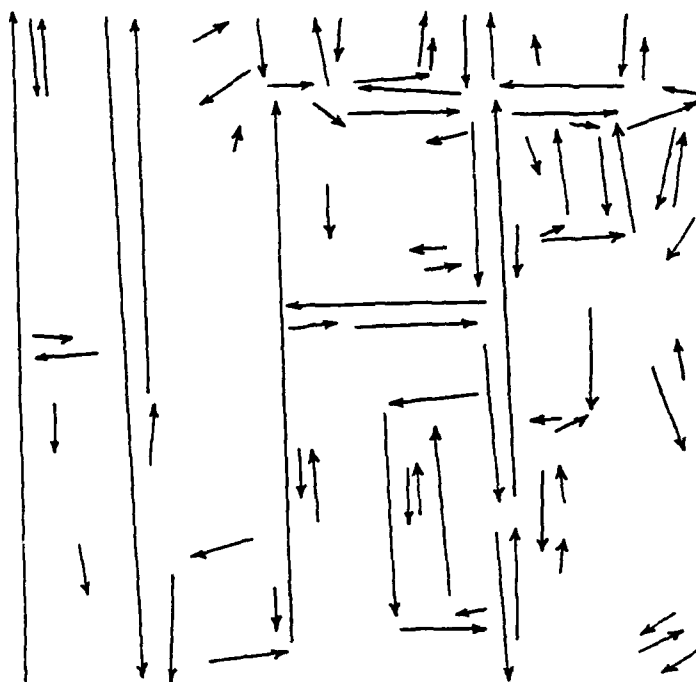
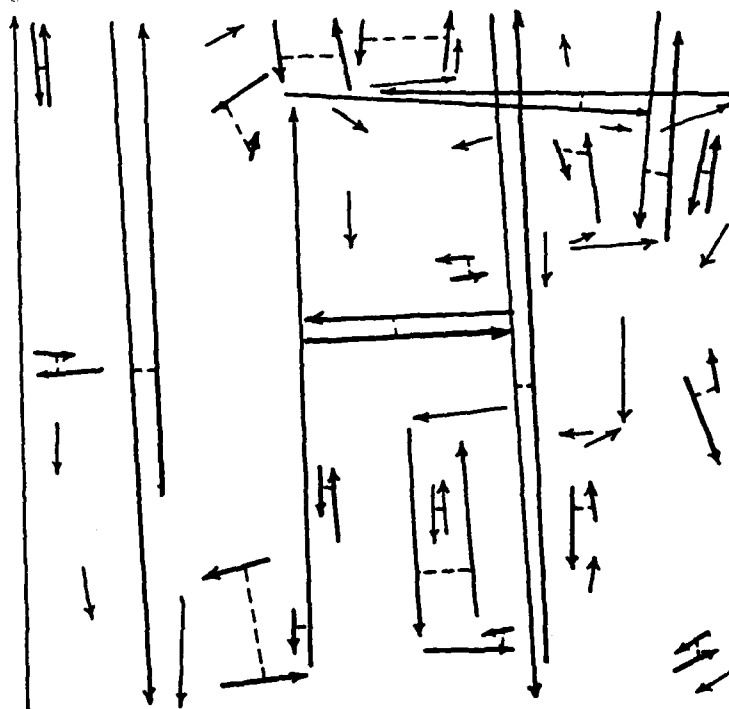


Figure 1a. Picture of part of an airfield.



b. Lines extracted from the picture in (a).



c. Result of applying the linking process. Dotted lines indicate linked pairs. The light lines are unpaired (noise) lines.

2. THE ALGORITHM

- 1) Read in the data for each line in the form $(x1, y1, x2, y2)$ where $(x1, y1)$ are the co-ordinates of the first endpoint and $(x2, y2)$ are the co-ordinates of the second endpoint of the line.
- 2) Initialize the links between each qualified pair of lines. (See below.)
- 3) Normalize each line's set of links and eliminate any line with no links.
- 4) Repeat steps 5-7 until no lines remain:
 - 5) For every link, say from line A to line B, and every line C that falls between A and B with its greatest link strength to line B, decrement the link from A to B by D's link strength to C if D falls outside A and B.
 - 6) Eliminate lines that are mutually best pairs and that have no other lines between them. Break the links from or to these lines.
 - 7) Re-normalize each line's set of link strengths and eliminate any line with no links to it or from it.

The algorithm starts by reading in the co-ordinates of the endpoints of each line. The slope of the line is defined by the order in which the endpoints appear. The length of the line is also calculated at this point.

For each line, all other lines that are anti-parallel to it and fall within a computed angle tolerance are examined to see if they are candidates for pairing with the line. The angle tolerance is computed as the sum of two factors $C1$ and $C2$, where $C1$ is a constant (currently 25 degrees) and $C2$ is inversely proportional to the longer of the two lines being considered. This provides greater laxity in interpreting short lines whose

directions are less certain than longer lines.

All lines which pass the angle test must in addition satisfy the further criterion that their overlap be non-zero. The overlap is computed as the intersection of the projections of the lines onto a line that runs between them and has an angle that is the mean of their angles modulo 180 degrees. (Figure 2.)

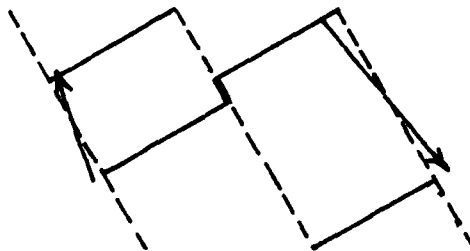


FIGURE 2

All lines passing the angle and overlap tests are linked, with a strength computed as a function of their distance from each other and their overlap. The distance between lines A and B is computed as the perpendicular distance between lines that pass through the midpoints of A and B, and whose direction is the average of those of A and B. (Figure 3.) The dotted lines in the figure have slope halfway between that of line A and line B.

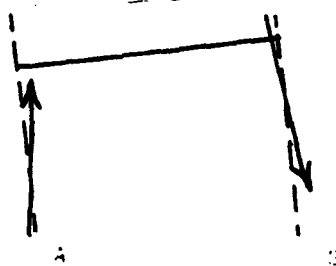


FIGURE 3

The current function being used for the link strength from line A to line B is

OVERLAP/LENGTH(LINE A)/DISTANCE

Several variants of these factors were used. Some of the variants that were tried are

1. $\text{strength} = (2 * \text{overlap} / (\text{length}(\text{line A}) + \text{length}(\text{line B})) / \text{distance}$
2. $\text{strength} = \text{overlap} / \text{length}(\text{line A}) + 1 / \text{distance}$
3. $\text{strength} = 2 * \text{overlap} / (\text{length}(\text{line A}) + \text{length}(\text{line B})) + 1 / \text{distance}$

Note that variants 1 and 3 are symmetric, while variant 2, and the variant described above are asymmetric. Asymmetric functions were preferred over symmetric functions because it seems reasonable that a short line should be more strongly attracted to a long line than the long line is to the short line. The algorithm proved remarkably insensitive to the initial link strength calculation; different schemes usually led to very similar results.

At this point, any lines without links are deleted. Next the iterative part of the algorithm is entered. Its goal is to find optimal pairings of line segments based on the initial estimates and satisfying the constraints that each line belong to at most one pair and that no two pairs overlap. (See Figure 4 where AB and CD are examples of mutual pairings which are ruled out by the overlap requirement).

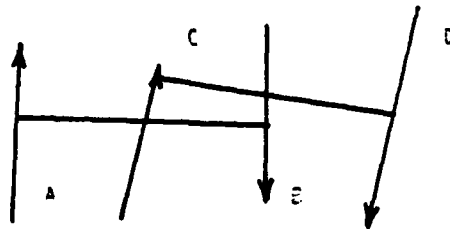


FIGURE 4

Each link, for instance from line A to line E, is updated in parallel according to the following scheme. If there is a line, say line C, between A and E, which is within the angle tolerance C' mentioned above, then C's greatest link strength is found, say to line D. If line D is also between A and E, nothing is done. Otherwise, A's link strength to E is decremented by C's link strength to D. After all the links have been updated in this manner, the link strengths are re-normalized. For all lines A and B, if A's greatest link strength is to B and B's greatest link strength is to A and there are no lines between A and B, then A and B are considered to be paired and are dropped from further consideration. At this point, any line with no links to it or from it is deleted, and a new iteration is started.

The between relation is calculated as follows (Figure 5.) To discover whether or not line C is between lines A and B, a line whose slope is the average of the slopes of lines A, B, and C is passed through the center points of each of these lines. (See the dotted lines in Figure 5.) Line C is between lines A and B if the intercept on some (arbitrary) axis of the dotted line through C's center is between the intercepts of the lines through A and B, and if line C overlaps both A and B.

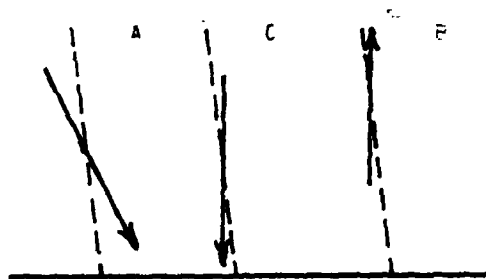


FIGURE 5

2. EXAMPLES AND DISCUSSION

The algorithm presented here performs competently on a variety of input data. This is especially significant when it is realized that no use is made of "holistic" patterns that might be apparent to humans who examine the data. Such patterns arise because of interpretations imposed on the scene that rely on assumptions external to the scene, and thus not available to the program. These assumptions vary from person to person, and different people often prefer different pairings in the same scene. As a result, it is not possible in general to specify how well the system performs unless there is an associated context for the interpretation.

2.1 Examples

In order to alleviate this problem, a series of digitized images was used as a source of lines for the program. The straight edges in the images were extracted using an iterative enhancement technique (Peleg, 1978), and the resulting lines were used as input to the pairing program. Figure 5 shows one such image and the set of lines extracted from it. The only contextual information retained from the images is the direction of each line. This direction is such that the image is lighter to the right of the line than to the left. It will be seen that even this small amount of knowledge can be valuable in disambiguating line pairings.

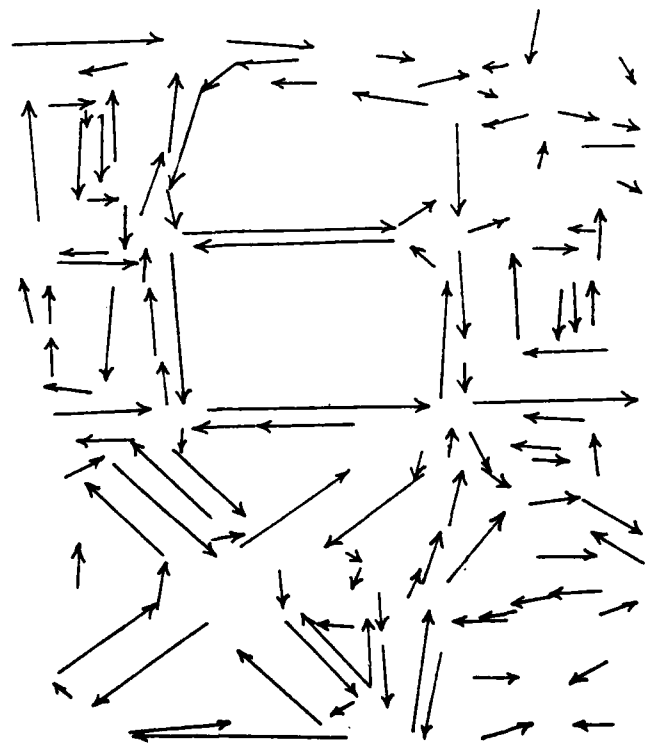
The use of images of real scenes enables an accurate evaluation of the pairings. It should, however, be clear that humans use much more information in forming pairs than is available to the program.

In the images that were used, good pairs correspond to the edges of roads and runways and to the sides of buildings. Many cultural features are characterized by antiparallel pairs of

Figure 6a. A picture of Lorton
Reformatory.



b. Lines extracted
from the picture in (a).



straight edges, and it is a useful step in interpreting the image to be able to discover these pairs. Because the edge extraction process is likely to break the lines into segments, and because of the existence of cross-roads, a program that attempts to link up collinear lines is used to extend and connect segments that lie on a straight line. This program will be the subject of another report. The following simple scheme was used to link collinear segments. A more ambitious scheme is described by Brooks (1973).

The algorithm starts by picking a line to work on. It then finds all other lines whose angle is sufficiently close to that of the chosen line. (Currently, within 25 degrees). These lines are the initial candidates for collinear linking. Most of the candidates are weeded out immediately by a constraint requiring centers of lines to lie within a narrow ribbon centered on the chosen line (usually 3 to 5 pixels wide.) (Figure 7).



FIGURE 7

It remains to link up lines that are not too far apart. This is essentially a parallel process. If the gap between the end point of one line and the start of the next is small enough (less than 1.5 times the maximum of the lines' lengths) the lines are merged. If not, the endpoint becomes the end of the linked segment, and the start point becomes the start of a new segment. Because the length that is used is not that of the whole new line, but only that of its final segment, the results do not depend on which line is chosen first. The pairing program receives as input the result of applying the collinearity program to the enhanced data.

Figure 8a shows a picture of a part of Lorton reformatory in the Fort Belvoir region of Virginia for which Figure 4b is the corresponding edge image. Figure 8a shows the result of applying the collinear algorithm, while Figure 8b shows the result of applying the program described in Section 2. Notice especially how the lines comprising the cross-shaped building in the lower left quadrant of the picture have been paired. The program has erroneously found the shadow borders, instead of those of the building. Figure 8c shows how a very small amount of contextual information can be useful in correcting this error. The extra information is that buildings usually appear brighter than their surroundings. It is manifested as a further constraint placed on the pairing process. This constraint is that a line may only pair with another, antiparallel, line if the intervening region is lighter than the area on the other side of the line. Edges are extracted with a direction corresponding to the convention that the lighter side of the edge be to the right. Thus, this constraint means that a line is only allowed to pair with other lines that are to its right. Using this constraint, the algorithm correctly discovers the borders of the building. Figure 9 shows another example of applying the program to data obtained from a suburban scene. In both cases, almost all pairs are correct, and most of the noise lines are not paired. The resultant description is much more useful for further processing than the original edge image.

3.2 Extensions

The pairing algorithm in this paper can be criticized for taking too local a view in determining the best matches. It might happen that a pair of lines that are mutually best partners should not, in fact, be linked. This can happen if the result of linking them is to cause other lines with low pairing strengths to be linked. This would give a low global pairing confidence, whereas forming alternative pairings might lead to a higher global

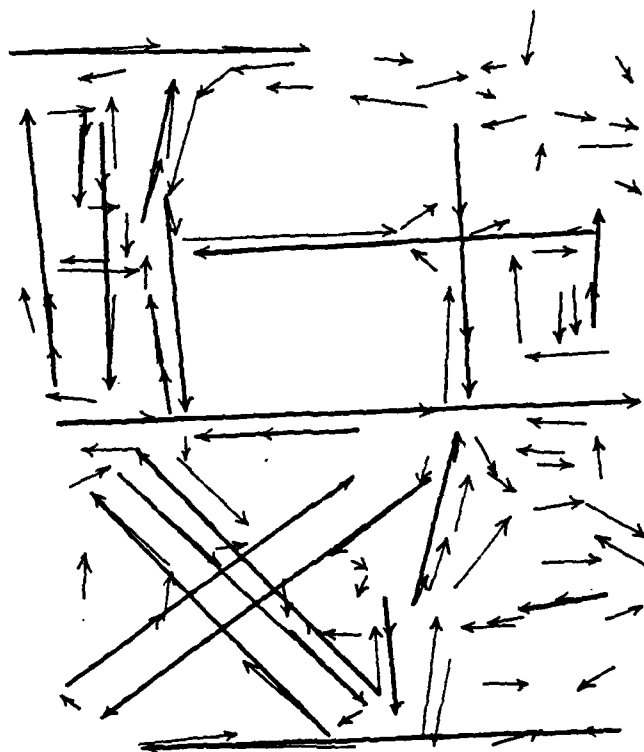
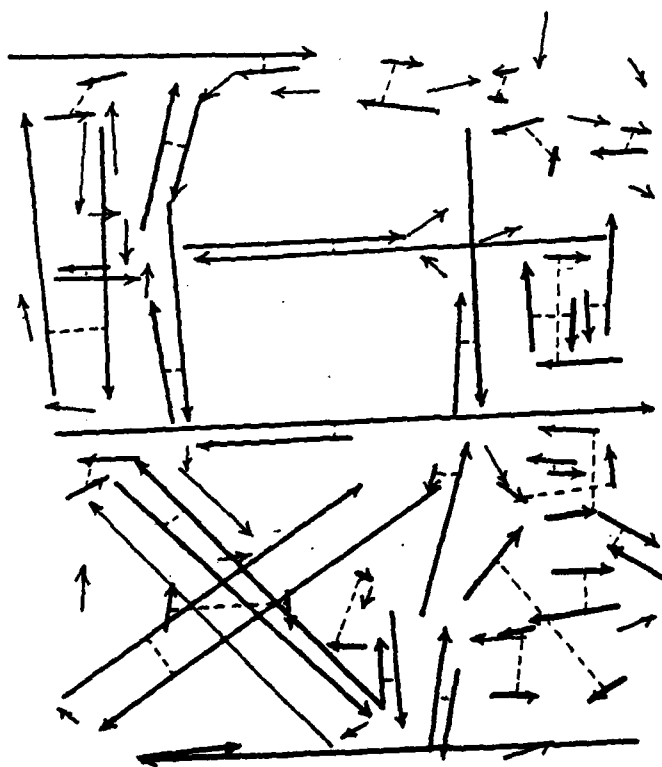
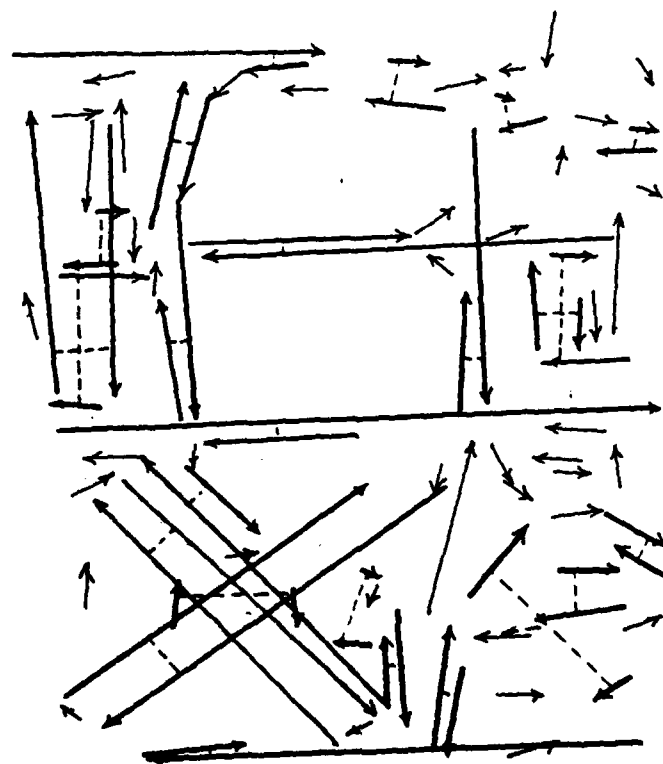


Figure 8a. Collinear linking of lines in Figure 6b.



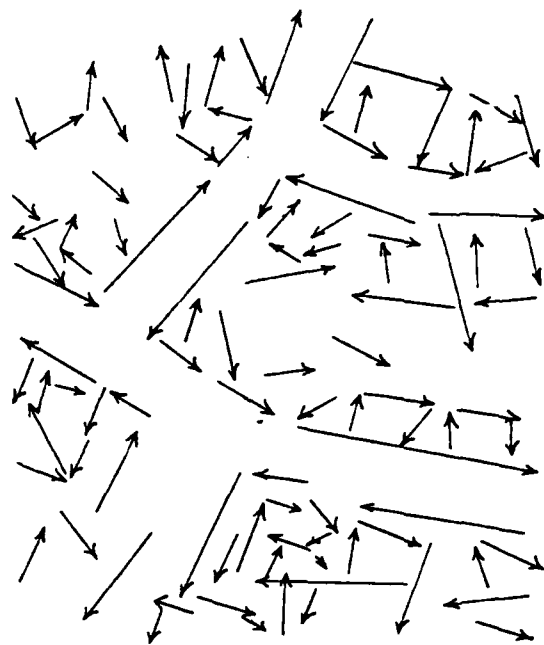
b. Naive anti-parallel linking applied to (b).



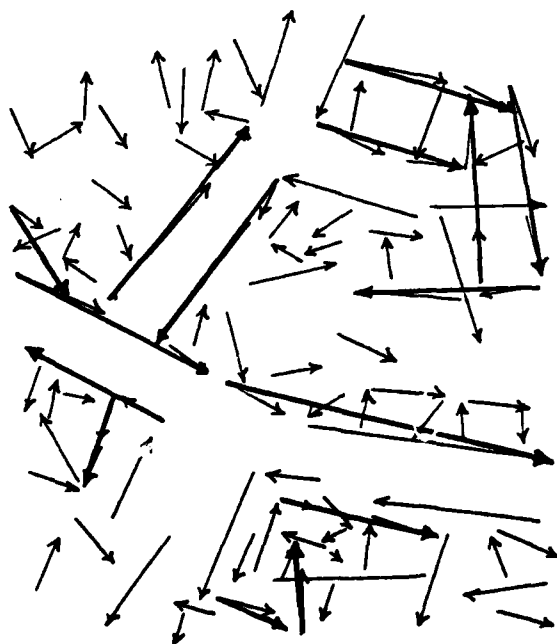
c. Anti-parallel linking with brightness constraint.



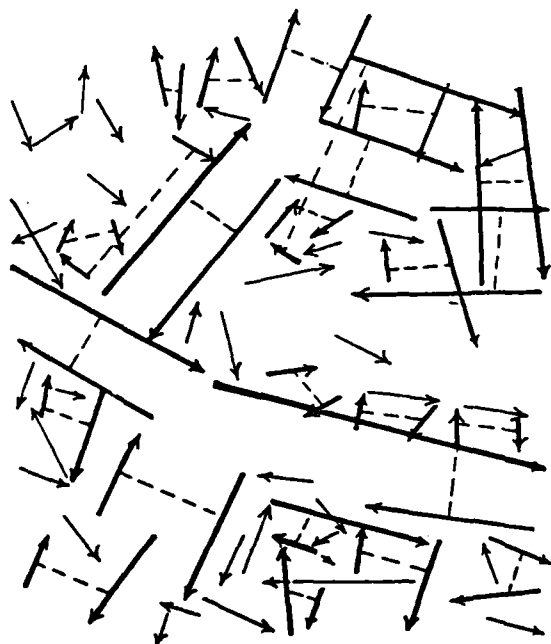
Figure 9a. A suburban scene.



b. Lines extracted from the picture in (a).



c. Collinear linking of lines in (b).



d. Anti-parallel linking.

confidence.

A way of alleviating this problem is to change the pairing process from a discrete process to a fuzzy one and to use a relaxation algorithm to find the best pairs. Instead of simply pairing lines that are mutually best links, and deleting all other links, it would be preferable to increase or decrease the confidence in each link according to the mutual compatibilities of the links with one another. By iterating this procedure, a more globally consistent result should emerge.

3.3 Alternatives

Several alternatives to the pairing procedure were considered, but were discarded in favor of the method described above. Two of the alternative methods are briefly described below.

The first alternative has the advantage of starting from a gradient or edge image, or even from a raw image, without having to do any preprocessing. It finds candidate antiparallel pairs of edges, but does not decide which are the best pairings.

The idea is to search out in opposite directions from each point in the image array, looking for edge responses. By requiring the responses to be at the same distance on either side of the point and to have opposite gradients, a measure of how close the point is to the axis of the antiparallel lines can be obtained. This measure should depend on the distance that has to be searched to find a pair of responses.

The result of applying this process at each point in the image, in parallel, is a set of responses that take the form of linear ridges centered between pairs of straight lines. These ridges encode the distance and overlap information for the surrounding lines, on a point-by-point basis. They can be used as input to a system similar to the current one.

The advantages of this method are that it extracts skeletons of those edges that are part of antiparallel pairs, and that it calculates properties of the pairs that are useful in (later)

processing. It suffers from flaws, however, which make it doubtful whether the method is as useful as the edge extraction and enhancement process that was actually used.

The main flaw is the inability of the method to find all possible pairs without further processing. Consider the situation in Figure 10 where the solid lines denote actual edges and the dotted lines denote axes of pairs.

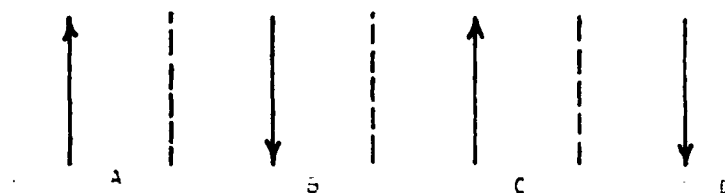


FIGURE 10

The axes correspond to the pairs AB, BC, and CD. However, the pair AD is also valid, but is not represented because its axis falls between lines B and C. By examining all axes that have the same slope, it is possible to discover the pairs that were initially missed, but this could be a costly process. Related work on shape skeletons (Wang, Wu and Rosenfeld, 1979) illustrates the difficulties of applying this kind of local processing.

The second alternative involves associating confidences with pairs based on similarities of features, and not just on relative size and position. Instead of calculating a merit function for each pair based on these features, a set of Hough-like transforms can be applied by projecting onto various subspaces of the feature space. This allows clusters of antiparallels to be detected, instead of having to deal with the lines pairwise.

Unfortunately, experience has shown that cluster detection in Hough spaces can be very difficult (Rosenfeld, 1979). In the linear feature domain, work by Broder et al. (in progress) seems to indicate a lack of significant clusters in Hough space. The

Hough-like method thus seems less promising than the method
advocated in this paper.

4. CONCLUSIONS

This paper has presented an algorithm for pairing antiparallel straight lines. The pairing was based on the amount of overlap between the lines, the distance between them, and on whether or not other lines were interposed.

The algorithm was evaluated using lines extracted from real images using an edge detection and enhancement process. The results were seen to be generally reasonable and the result of applying the process was a new image representation that was more useful for further processing than the initial edge image.

REFERENCES

1. Broder, et al. (in progress)
2. A. W. Brooks, Local directed edge linking and ribbon finding. Proc. DARPA Image Understanding Workshop, SRI International, Menlo Park, CA, April 1977, pp 72-78.
3. N. Levatia and K. P. Sabu, Linear feature extraction. Proc. DARPA Image Understanding Workshop, Carnegie-Mellon University, Pittsburgh, PA, November 1978, pp 73-75.
4. S. Peleg and A. Rosenfeld, Straight edge enhancement and mapping. TR-394, Computer Science Center, University of Maryland, College Park, MD, September 1978.
5. L. H. Guam, Road tracking and anomaly detection in aerial imagery, Proc. DARPA Image Understanding Workshop, MIT, Cambridge, MA, May 1978, pp 51-55.
6. A. Rosenfeld, Levels of representation in cultural feature extraction. Proc. DARPA Image Understanding Workshop, USC, Los Angeles, CA, November 1979, pp 112-127.
7. Tavakoli (in progress)
8. S. Wang, A. Rosenfeld, and W. Wu, A Medial Axis Transformation For Grayscale Pictures. TR-843, Computer Science Center, University of Maryland, College Park, MD, December 1979.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
	AD-A086 086	
4. TITLE (and Subtitle) A METHOD FOR FINDING PAIRS OF ANTI- PARALLEL STRAIGHT LINES		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER TR-845 ✓
7. AUTHOR(s) Ann Scher Michael Shneier Azriel Rosenfeld		8. CONTRACT OR GRANT NUMBER(s) DAAG-53-76C-0138
9. PERFORMING ORGANIZATION NAME AND ADDRESS Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, MD 20742		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Night Vision Laboratory Fort Belvoir, VA 22060		12. REPORT DATE December 1979
		13. NUMBER OF PAGES 20
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Image processing Pattern recognition Parallel lines		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A method of pairing anti-parallel straight lines is presented and discussed. The pairing is based on the distance between the lines, the amount by which they overlap, and on whether or not other lines are interposed. Examples are shown of applying the method to high-resolution aerial photographs. Results indicate that cultural features such as roads and buildings can be extracted and that a significant reduction in the complexity of the image description can be obtained.		

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)